Machine Learning Based Lightpath Classifier for Impairment Aware Resource Allocation in SDM-EONs

Jaya Lakshmi Ravipudi
Charles L. Brown Department
of Electrical and Computer Engineering
University of Virginia
Charlottesville, VA-22904, USA
jr3vz@virginia.edu

Maïté Brandt-Pearce
Charles L. Brown Department
of Electrical and Computer Engineering
University of Virginia
Charlottesville, VA-22904, USA
mb-p@virginia.edu

***Abstract*—Efficient resource allocation is needed for next-generation space division multiplexing-based elastic optical networks (SDM-EON). This paper presents a deep neural network (DNN) classifier for dynamic routing and core selection in SDM-EONs. It uses an advanced optimization algorithm to predict lightpath availability, considering the quality of transmission and resource availability. The DNN training accounts for spectral availability, and linear and nonlinear physical layer impairments experienced in SDM-EONs. In addition, a feature importance analysis is performed for different traffic loads. The DNN performance and blocking probability are shown to be superior by lowering blocking by more than a factor of 2 compared with the benchmark techniques, when tested on the 14-node NSFNET topology with unseen network states.**

***Index Terms*—SDM-EONs, deep neural networks, resource allocation, physical layer impairments**

I. Introduction

Elastic optical networks (EONs) have been widely researched over the last decade due to their better spectrum utilization than wavelength-division multiplexed optical networks. However, the capacity of currently deployed single-core EONs is nearly exhausted, and multi-core fibers (MCFs) have been proposed as a solution [1]. The spatially separated cores of MCFs increase the capacity and enable the creation of a space-division multiplexed elastic optical network (SDM-EON).

A traffic request can get blocked in the optical network under two scenarios: first, if requested resources that obey the EON's spectral continuity and contiguity constraints are unavailable, and second, if the resulting quality of transmission (QoT) is unsatisfactory or causes degradation of existing signals in operation. Physical layer impairments (PLIs) such as amplified spontaneous emission (ASE), crosstalk (XT), self-phase modulation (SPM), and cross-phase modulation (XPM) degrade the signal QoT and limit the transmission reach. Hence, QoT-aware lightpath suitability prediction is necessary for effective planning of the optical network.

In recent literature, PLIs are often estimated using the Gaussian noise (GN) model or its improved versions [2]–[4]. The GN model is an approximate formula that saves computation time compared to the split-step Fourier method. However, computing the QoT using the GN model for each candidate spectral block in the spectrum assignment phase can still be overly time-consuming [5]. Another approach is machine learning (ML)-based QoT estimation. Recent works on ML-aided QoT estimation use techniques such as K-nearest neighbors, random forests, neural networks, and DNNs [6]–[11].

However, all these works consider single-core fibers; to our knowledge, no published works on ML-enabled optical networking for multicore fibers consider all major impairments listed above. The previously published ML algorithms estimate or classify the lightpath solely on the QoT compared to a QoT metric threshold, ignoring the question of lightpath unavailability due to spectral resource shortages. Moreover, the ML optical networking community has largely focussed on improving the ML models only by enriching the training dataset quality but not on improving the inner workings of the ML models. Advanced training algorithms, as used in this work, can aid current ML efforts to improve the prediction capabilities of the models. In addition, the previously proposed ML models have been well analyzed for a given traffic load, but no work has considered all impairments under different traffic load scenarios for model robustness.

This paper proposes a deep neural network (DNN) approach to resource allocation for SDM-EON to ensure the effective use of the multi-core network and the following contributions are made keeping these research gaps in mind:

- A DNN-based binary classifier is proposed for impairment-affected SDM-EONs. All impairments (ASE, XPM, SPM, and XT) are considered for realistic results.
- We use ML in a novel way to predict the acceptability of a lightpath, taking into account the QoT and the availability of end-to-end spectral resources.
- An advanced optimization algorithm called the Jaya algorithm [12] is presented for DNN training in the optical

networking domain for the first time.
- Varying traffic loads are considered for improving prediction capability and robustness, and a feature importance analysis is provided.
- The proposed models are expected to aid PLI-aware resource allocation algorithms; hence, a preliminary integration of the proposed model into a routing, modulation, spectrum, and core allocation (RMSCA) application is presented that shows reduced blocking by more than a factor of 2 when tested on the 14-node NSFNET topology.

Section II describes the proposed classifier followed by a description of the Jaya algorithm in Section III. Section IV presents the classifer performance and feature importance results. Section V presents a preliminary application of the DNN model to RMSCA followed by conclusions in Section VI.

II. Proposed Binary Classifier

A. Dataset generation

Public datasets for optical networks such as Abilene, GEANT, and Mendeley Data [13] do not pertain to SDM-EONs, and hence we generate synthetic data as is commonly followed in similar studies [14]–[16]. The datasets are generated on the 14-node NSFNET with 7 cores per link for 200 to 600 Erlangs of dynamic random-bandwidth traffic [17]. The resource allocation algorithm uses K-shortest paths (KSP) for routing, QoT-guaranteed spectrally efficient modulation formats [18], first core for core selection, and first fit (FF) for spectrum allocation under XT, ASE, SPM, and XPM impairments. The SDM-EON system parameters and impairment models are the same as used in our previous work [19, Table 7]. The number of requests, R, considered are $10,000$ and $100,000$ with data rates uniformly distributed in $[50, 600]$ Gbps. The dataset is appended with the results of the resource provisoning for all possible candidates for a given request. This method ensures a diverse and extensive collection of samples, avoiding information wastage about possible route/core/spectrum combinations instead of just including the successfully provisioned combination.

B. Classifier performance metric

The classification considered in this work is of an imbalanced type, i.e., the occurrence of the two classes (1 and 0) in the generated datasets is not in equal proportion. This is expected since rejections are rare at low loads and increase with traffic. The widely used classification metric, which is to measure the proportion of correct decisions, fails for imbalanced problems, as the accuracy in predicting a 1 will always be better if the dataset has a major proportion of class 1 samples [20]. Hence, we instead evaluate our classifier using the area under the receiver operating characteristic curve (AUC ROC) metric [7], which lies in $[0, 1]$. In general, any classifier with an AUC ROC ≥ 0.8 is considered very good, and ≥ 0.9 is deemed excellent for imbalanced problems [20] [21]. An ideal classifier has an AUC ROC $= 1.0$.

C. Dataset features

We consider eight lightpath features, f_{pn}: required spectrum slots, modulation format, path length, core spectrum usage, core fragmentation, number of continuity and contiguity following spectrum blocks, number of occupied slots in adjacent cores, and number of active spectrum blocks on the current core. The output label is a 1 or 0 denoting the acceptability or rejection of the lightpath, respectively.

The values of the $N = 8$ f_{pn} on a given lightpath L and core c are computed as below. These features are collected as part of dataset generation and, during the RMSCA application, computed as the input feature vector using the dynamic network state.

1) Number of frequency slots required for the request with 1 slot as guardband:
$$S_{req} = \frac{\text{Datarate}}{2 * \delta f * \eta_m} + 1, \tag{1}$$
where δf is the frequency granularity of the EON and η_m is the spectral efficiency of modulation format $m \in \{$BPSK, QPSK, 8-QAM,16-QAM, 32-QAM$\}$.
2) Modulation format m, chosen using the QoT model from [18].
3) Path length l_L for lightpath L:
$$l_L = \sum_{e \in L} \ell^{(e)}, \tag{2}$$
where $\ell^{(e)}$ is the length of fiber link e.
4) Spectrum usage of core c over lightpath L:
$$S_L^{(c)} = \sum_{e \in L} S^{(c,e)}, \tag{3}$$
where $S^{(c,e)}$ is the number of occupied slots on core c of link e.
5) Fragmentation of core c on the entire lightpath L:
$$F_L^{(c)} = \sum_{e \in L} F^{(c,e)}, \tag{4}$$
where $F^{(c,e)}$ is the fragmentation on core c of link e, given by [22]
$$F^{(c,e)} = \sum_{\gamma_{ec} \in \Gamma_{ec}} \frac{|\gamma_{ec}|}{S} \cdot \ln \frac{S}{|\gamma_{ec}|}. \tag{5}$$
Γ_{ec} is the set of fragments on link e of core c, $|\gamma_{ec}|$ is the size of fragment γ_{ec}, i.e., number of frequency slots in that fragment, and S is the total number of frequency slots. This fragmentation metric is based on the entropy concept and better captures the difference in fragment sizes compared to the general external fragmentation formula [22].
6) Number of spectrum blocks of size S_{req} on core c that obey the continuity and contiguity constraints over lightpath L, denoted as $\mathbb{B}_L^{(c)}$.
7) Number of active spectrum blocks on core c, denoted as $B_L^{(c)}$.

8) Number of occupied slots in adjacent cores, denoted as $S_L^{(adj)}$.

Given this set of features, the DNN classifier outputs a prediction value given by

$$Y = \text{Classifier}(S_{req}, m, l_L, S_L^{(c)}, F_L^{(c)}, \mathbb{B}_L^{(c)}, B_L^{(c)}, S_L^{(adj)}) \quad (6)$$

In the training stage, the prediction is compared with the hard label (0 or 1) and the neural network paramters are updated accordingly. In a RMSCA implementation, the prediction values from different set of candidates can be utlized to make soft decisions (as shown in this work) and/or hard decisions in terms of 0 or 1.

It may be noted that there are many features to choose from when considering how to best capture the underlying network scenario for predictions aligned with the goal of the problem. In this work, initially, 13 relevant features were chosen, including the source, destination, number of traversed links, maximum link length and the datarate, in addition to the chosen 8 features described above. An exhaustive list of features can be found in [23]. However, the prediction capability of the classifier reduced with the full set of 13 features as shown in Fig. 1 and also when any of their combinations were chosen. Hence, the 8 feature set is chosen in this work, reducing the computational cost compared to the full set.

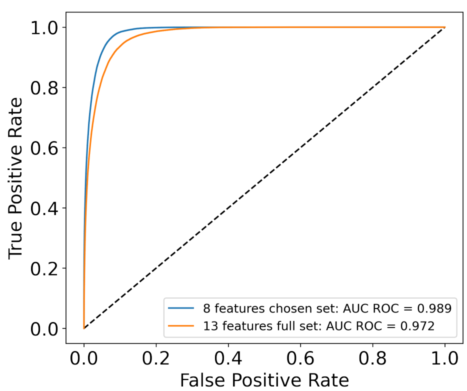

Fig. 1. Difference in classifier quality when using the full 13 features vs. the chosen 8 features, at a load of 300 Erlangs

The DNN performance is affected by various hyperparameters [24]. The various combinations tested and the values chosen that resulted in the best performance are given in Table I. In particular, the last hidden layer with 8 neurons also performed competitively with 16 neurons. However, the 16 neurons' structure performed slightly better, especially for higher loads, so it was chosen for the final model for all loads.

III. Jaya Algorithm for DNN Training

The aim of DNN training is to learn the optimal set of weights and biases that minimizes the loss function of the DNN, i.e., the binary cross entropy (BCE), for our case [26]. DNN training algorithms differ in scalability, interpretability, and sensitivity to outliers. Commonly used training algorithms

TABLE I
Hyperparameters and Their Tested Ranges

Hyperparameter	**Tested Values**	**Chosen Value**
Learning rate	10^{-1} to 10^{-5}	10^{-4}
Number of hidden layers	$2, 3, 4$	3
Number of neurons per hidden layer	$16, 32, 64$	32
Number of neurons in last hidden layer	$2, 4, 8, 16, 32, 64$	16
Activation functions [25]	Sigmoid, ReLU, Tanh	ReLU, Sigmoid
Batch size	$32, 64, 128, 256, 1024$	256
Regularizer	l_1, l_2	l_2
Initializer	Glorot, He normal	He normal

such as Adam, stochastic gradient descent, RMSProp, and Adagrad [25] are gradient-based. Metaheuristic algorithms can be used when the optimization problem is non-convex or high dimensional and when gradient-based algorithms cannot find a good solution. Examples include particle swarm optimization (PSO), genetic algorithms (GA), ant colony optimization (ACO), differential evolution (DE), etc. [27]

In this paper, we use an advanced metaheuristic algorithm called the Jaya algorithm [12] for DNN training and compare the performance with Adam. The Jaya algorithm has been shown to perform better than the commonly used optimizers in various problem domains [27] [28]. Algorithms such as PSO, GA, DE, ACO, etc., have their algorithm-specific parameters that must be tuned before applying them to the optimization problem. In contrast, the simplicity of the Jaya algorithm lies in the absence of such parameters; only the common control parameters, such as population size and termination criteria, need to be defined [27] [28]. Fig. 3 shows a flowchart of the Jaya algorithm applied for DNN training, and the detailed working of the algorithm can be found in [12]. We show only the relevant details here to understand the current work.

The Jaya optimization algorithm operates using a candidate population of size U, a number of variables V, and runs until the termination criteria are reached, which in this paper is taken as $|I|$, the number of iterations. In our case, the variables are the DNN weights and biases whose values we want to find. The population refers to the number of candidate solutions (set of variable values), which are randomly generated at the beginning of the algorithm. The values are updated using

$$\begin{aligned} X_{u,v,i+1} = & X_{u,v,i} + r_{1,u,i}(X_{u,\text{best},i} - |X_{u,v,i}|) \\ & + r_{2,u,i}(X_{u,\text{worst},i} - |X_{u,v,i}|) \end{aligned} \quad (7)$$

where $u = 1, \cdots, U$, $v = 1, \cdots, V$, $i \in I$ denote the candidate, variable and iteration number, respectively. The new solution $X_{u,v,i+1}$ is computed as a function of the value $X_{u,v,i}$ from the previous iteration, the best and worst values of that variable, denoted as $X_{u,\text{best},i}$ and $X_{u,\text{worst},i}$, respectively, and two random numbers, $r_{1,u,i}$ and $r_{2,u,i}$, in the range $(0, 1)$. The best and worst solutions for a minimization problem are the minimum and maximum values of the variable at the previous iteration based on the optimization cost function. The population values, i.e., the DNN weights and biases, are updated at each iteration using (7).

The convergence plot of the classifier is shown in Fig. 2. The parameters of Adam are given in Table I and the Jaya algorithm uses a population size of 50. The training loss is computed using the binary cross-entropy loss [26].

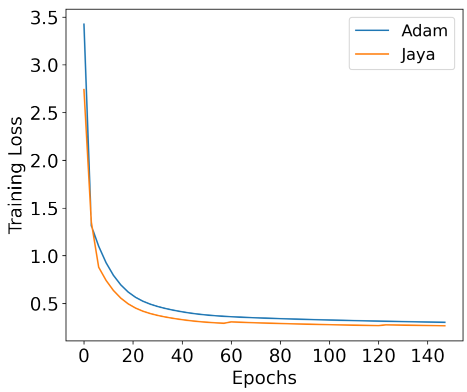

Fig. 2. Convergence plot of the proposed classifier for Adam and Jaya optimizers

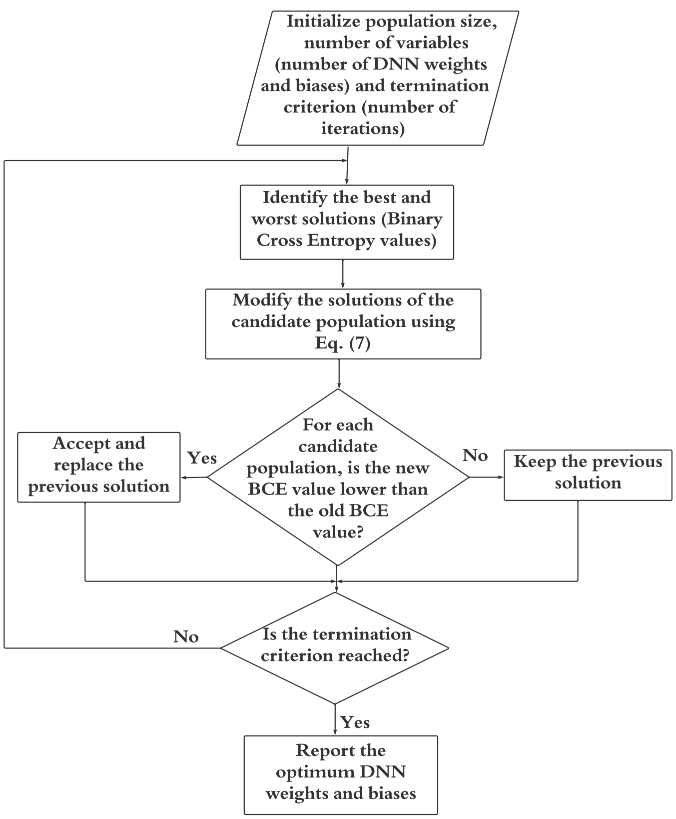

Fig. 3. Flowchart of the Jaya algorithm applied for DNN training

IV. Results and Analysis

In practice, a query is made to the DNN, and it identifies whether the lightpath will be viable or not. The classification task over varying traffic loads is assumed to be solvable using two approaches: we denote as *Method 1* when a single DNN is used for all loads, and call *Method 2* when a separately trained DNN is employed for each load. To compare Method 1 and Method 2, we train the DNN on 80% of the data and test the AUC ROC performance on the remaining 20% unseen data (reserved for testing), shown in Table II. The methods are compared for generated datasets with $R = 10,000$ and $R = 100,000$ per traffic load for robustness. Method 2 obtains higher AUC ROCs (in bold) for all the tested loads and for both values of R. Hence, we use Method 2 for the remaining results.

TABLE II
Comparison of Testing AUC ROC Obtained with Methods 1 and 2 using the Adam Optimizer

	$R = 10,000$		$R = 100,000$	
Load	**Method 1**	**Method 2**	**Method 1**	**Method 2**
200	0.981	**0.987**	0.979	**0.982**
300	0.975	**0.981**	0.966	**0.974**
400	0.964	**0.973**	0.958	**0.963**
500	0.956	**0.965**	0.947	**0.953**
600	0.943	**0.952**	0.932	**0.941**

We now compare the performance of training algorithms on the AUC ROC using the 20% unseen test data for the $R = 100,000$ dataset. Fig. 4 shows the difference between the actual AUC ROC observed from simulations and the ideal AUC ROC $= 1.0$ for each DNN trained at each load using Adam and Jaya. The Jaya-trained DNNs are closer to an ideal classifier than the Adam-trained DNNs for all loads. Hence, the Jaya-trained classifiers would perform better in practice when unseen EON data is input to the DNN. This is because the Jaya algorithm is able to minimize the binary cross entropy loss better than Adam, and the resulting weights and biases provide comparatively better classification capability at each load.

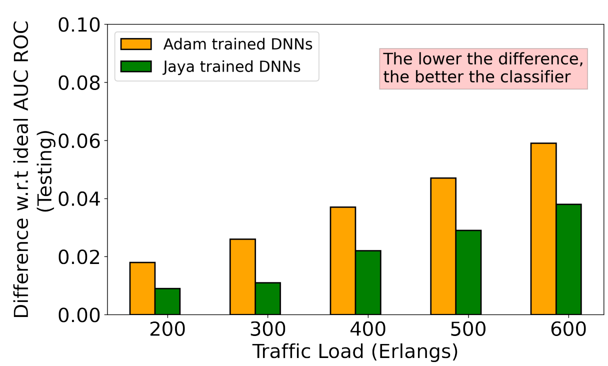

Fig. 4. Difference between simulated AUC ROC and the ideal AUC ROC $=$ 1.0 using Adam vs. Jaya trained DNNs (Method 2) for $R = 100,000$

Method 2 defined above presumes that the operational network load is one of the loads the DNN was trained for. It is thus worthwhile to investigate its robustness when this assumption is violated. Additional datasets for 350 and 525 Erlangs are generated and tested using DNNs trained on other loads. Table III shows the AUC ROCs when the actual network load is 350 and 525 Erlangs. For both loads, DNNs trained on a higher load, i.e., 400 and 600, respectively, performed better. When the DNN was trained at 300 and 500 Erlangs, the lower AUC ROC values can be attributed to an underprediction of QoT failures for loads 350 and 525 Erlangs, respectively. Similar reasoning follows for the Jaya trained DNNs at 400 and 600 Erlangs but it again has a better performance (in bold) than the Adam trained ones.

TABLE III
TEST AUC ROC FOR UNTRAINED LOADS USING ADAM VS. JAYA TRAINED DNNS (METHOD 2) FOR $R = 100,000$

Actual Load	Trained Load	Adam	Jaya
350	300	0.921	0.937
	400	0.952	**0.968**
525	500	0.912	0.926
	600	0.928	**0.949**

Which features are considered for training impacts the classifier performance; sometimes only a subset of the features is enough. Table IV shows the effect of removing one feature at a time on Jaya-trained DNNs for loads 300 and 600 Erlangs. The modulation format, m and S_{req} remain the most important features in determining the lightpath suitability as their removal lowers the AUC ROC the most, i.e., by $4-6\%$ and $6-8\%$ for 300 and 600 Erlangs, respectively, compared to the scenario when all features are present. As the load increases to 600 Erlangs, the $B_L^{(c)}$ and l_L features becomes more important. The percentage decrease for every missing feature affirms that the set of eight features considered in this work is crucial for the proposed DNN classifier.

TABLE IV
EFFECT OF DIFFERENT FEATURES ON PREDICTION AUC ROC FOR LOADS 300 AND 600 ERLANGS USING THE JAYA OPTIMIZER

Absent feature	Prediction AUC ROC	
	300 Erlangs	600 Erlangs
m	0.945	0.903
S_{req}	0.951	0.887
$S_L^{(adj)}$	0.961	0.939
$F_L^{(c)}$	0.962	0.933
$B_L^{(c)}$	0.968	0.911
$\mathbb{B}_L^{(c)}$	0.974	0.947
$S_L^{(c)}$	0.976	0.959
l_L	0.981	0.922
All features present	**0.989**	**0.962**

V. APPLICATION OF THE PROPOSED CLASSIFIER TO RMSCA

In this section, we show the application of our proposed DNN lightpath classifier to QoT-aware dynamic RMSCA, where the classifier encounters unseen network states. The conventional K-shortest path (KSP) routing subjected to the QoT constraint is used as a benchmark for a fair comparison with the proposed algorithm. Our RMSCA approach is described in Algorithm 1, which takes as input the K number of shortest paths for the current request, the set of cores (C) available, and the dynamic network state. The total number of candidates is $K \times |C|$. The DNN predictions on the path-core candidates are used as their scores, indicating their quality. Since we use a binary classifier, a threshold (V_{th}) is initially used to filter out the candidates the classifier considers suitable. The best V_{th} ((8)) for the DNN of each load is found using the geometric mean (GM) technique [29]. The GM for an imbalanced classifier seeks a balance between sensitivity and specificity and is a point on the AUC ROC curve that maximizes the 0/1 distinguishing capability. The maximum GM is computed as

$$\max_i(\sqrt{TPR_i * (1 - FPR_i)}) \tag{8}$$

where i is a threshold point on the AUC ROC curve, TPR_i is the true positive rate at point i, and FPR_i is the false positive rates at point i.

The ranked candidates are used for subsequent spectrum assignments. The spectrum assignment step finds a spectral block using the FF algorithm that satisfies the continuity, contiguity, and QoT conditions on the particular route-core combination.

Algorithm 1 Path and core selection

procedure DNN_ROUTE_CORE(Network topology, K shortest paths, set of cores C, network state)
    for each candidate in set of $K \times |C|$ candidates **do**
        Form the feature vector
        Predict the candidate's suitability using the trained DNN model of the corresponding traffic load
        if prediction $\leq V_{th}$ **then**
            Store the prediction
        end if
    end for
    Sort the predictions in ascending order and rearrange the corresponding $K \times |C|$ candidates
    return Ranked $K \times |C|$ candidates to the RMSCA procedure
end procedure

Fig. 5 shows the blocking probability obtained using the benchmark KSP (using first core selection), labeled KSP in the legend, and DNN-enabled path core selection, both under FF spectrum allocation. Results are drawn with 95% confidence intervals marked as error bars around each mean value over five tests. Fig. 5 shows that the proposed DNN approach obtains a significantly lower blocking probability (more than a factor of two) than the conventional KSP at all loads considered. Hence, the performance indicates that the proposed DNN model can accurately predict the lightpath suitability using key network features. The superior performance of DNN-enabled RMSCA is because it assesses all the viable candidates through their DNN prediction values and selects the best one; however, this improvement comes at the cost of collecting features for all candidate routes and core combinations, which increases the computational cost.

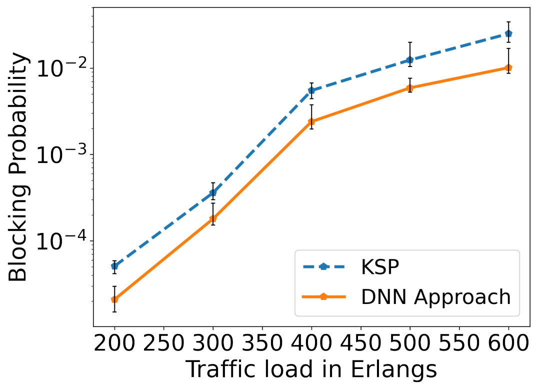

Fig. 5. Blocking probabilities for the KSP and proposed DNN approach in a 7-core NSFNET

VI. Conclusions

A DNN-based binary classifier using a recently published optimization algorithm known as the Jaya algorithm is proposed for the first time to predict the lightpath suitability for impairment-aware resource allocation in SDM-EONs. Both QoT-blocking and spectrum availability are included in the definition of an unsuitable lightpath. All major physical layer impairments (ASE, XT, SPM, and XPM) are included in our model. The Jaya-based classifier predicts lightpath suitability better than the conventional Adam optimizer for varying loads and unseen network states. The impact of load, training size, and feature set are analyzed. The Jaya-enabled binary classifier can improve PLI-aware resource allocation in SDM-EONs and offer lower blocking probability than conventional RMSCA approaches.

References

[1] P. Bayvel, R. Maher, T. Xu, G. Liga, N. A. Shevchenko, D. Lavery, A. Alvarado, and R. I. Killey, "Maximizing the optical network capacity," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2062, p. 20140440, 2016.

[2] P. Poggiolini, "The GN model of non-linear propagation in uncompensated coherent optical systems," *Journal of Lightwave Technology*, vol. 30, no. 24, pp. 3857–3879, 2012.

[3] A. Carena, G. Bosco, V. Curri, Y. Jiang, P. Poggiolini, and F. Forghieri, "EGN model of non-linear fiber propagation," *Optics express*, vol. 22, no. 13, pp. 16 335–16 362, 2014.

[4] D. Semrau, R. I. Killey, and P. Bayvel, "The Gaussian noise model in the presence of inter-channel stimulated Raman scattering," *Journal of Lightwave Technology*, vol. 36, no. 14, pp. 3046–3055, 2018.

[5] P. Poggiolini and Y. Jiang, "Recent advances in the modeling of the impact of nonlinear fiber propagation effects on uncompensated coherent transmission systems," *Journal of Lightwave Technology*, vol. 35, no. 3, pp. 458–480, 2016.

[6] S. Allogba, S. Aladin, and C. Tremblay, "Machine-learning-based lightpath QoT estimation and forecasting," *Journal of Lightwave Technology*, vol. 40, no. 10, pp. 3115–3127, 2022.

[7] C. Rottondi, L. Barletta, A. Giusti, and M. Tornatore, "Machine-learning method for quality of transmission prediction of unestablished lightpaths," *Journal of Optical Communications and Networking*, vol. 10, no. 2, pp. A286–A297, 2018.

[8] I. Khan, M. Bilal, M. U. Masood, A. D'Amico, and V. Curri, "Lightpath QoT computation in optical networks assisted by transfer learning," *Journal of Optical Communications and Networking*, vol. 13, no. 4, pp. B72–B82, 2021.

[9] J. Lu, Q. Fan, G. Zhou, L. Lu, C. Yu, A. P. T. Lau, and C. Lu, "Automated training dataset collection system design for machine learning application in optical networks: an example of quality of transmission estimation," *Journal of Optical Communications and Networking*, vol. 13, no. 11, pp. 289–300, 2021.

[10] S. Petale and S. Subramaniam, "An ML approach for crosstalk-aware modulation format selection in SDM-EONs," in *2022 International Conference on Optical Network Design and Modeling (ONDM)*. IEEE, 2022, pp. 1–6.

[11] M. Salani, C. Rottondi, and M. Tornatore, "Routing and spectrum assignment integrating machine-learning-based QoT estimation in elastic optical networks," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1738–1746.

[12] R. V. Rao, *Jaya: an advanced optimization algorithm and its engineering applications*. Springer, 2019.

[13] T. Panayiotou, M. Michalopoulou, and G. Ellinas, "Survey on machine learning for traffic-driven service provisioning in optical networks," *IEEE Communications Surveys & Tutorials*, 2023.

[14] H. Yang, Q. Yao, B. Bao, A. Yu, J. Zhang, and A. V. Vasilakos, "Multi-associated parameters aggregation-based routing and resources allocation in multi-core elastic optical networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 5, pp. 2145–2157, 2022.

[15] J. Müller, S. K. Patri, T. Fehenberger, H. Griesser, J.-P. Elbers, and C. Mas-Machuca, "QoT estimation using EGN-assisted machine learning for multi-period network planning," *Journal of Optical Communications and Networking*, vol. 14, no. 12, pp. 1010–1019, 2022.

[16] R. A. Oliveira, D. Rosário, E. Cerqueira, and H. Oliveira, "Machine learning assisted traffic-aware approach to path assignment in SDM-EONs," in *Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC, 2022, pp. 29–42.

[17] J. L. Ravipudi and M. Brandt-Pearce, "A score function heuristic for crosstalk-and fragmentation-aware dynamic routing, modulation, core, and spectrum allocation in SDM-EONs," in *2022 IEEE Future Networks World Forum (FNWF)*. IEEE, 2022, pp. 83–87.

[18] L. Yan, E. Agrell, H. Wymeersch, and M. Brandt-Pearce, "Resource allocation for flexible-grid optical networks with nonlinear channel model," *Journal of Optical Communications and Networking*, vol. 7, no. 11, pp. B101–B108, 2015.

[19] J. L. Ravipudi and M. Brandt-Pearce, "Impairment-and fragmentation-aware dynamic routing, modulation and spectrum allocation in C+L band elastic optical networks using Q-learning," *Optical Switching and Networking*, vol. 47, p. 100717, 2023.

[20] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.

[21] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013, vol. 398.

[22] P. Lechowicz, M. Tornatore, A. Włodarczyk, and K. Walkowiak, "Fragmentation metrics and fragmentation-aware algorithm for spectrally/spatially flexible optical networks," *Journal of Optical Communications and Networking*, vol. 12, no. 5, pp. 133–145, 2020.

[23] O. Ayoub, S. Troia, D. Andreoletti, A. Bianco, M. Tornatore, S. Giordano, and C. Rottondi, "Towards explainable artificial intelligence in optical networks: the use case of lightpath qot estimation," *Journal of Optical Communications and Networking*, vol. 15, no. 1, pp. A26–A38, 2022.

[24] L. Owen, *Hyperparameter Tuning with Python: Boost Your Machine Learning Model's Performance Via Hyperparameter Tuning*. Packt Publishing Limited, 2022.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016, http://www.deeplearningbook.org.

[26] A. Bhardwaj, W. Di, and J. Wei, *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. Packt Publishing Ltd, 2018.

[27] R. A. Zitar, M. A. Al-Betar, M. A. Awadallah, I. A. Doush, and K. Assaleh, "An intensive and comprehensive overview of JAYA algorithm, its versions and applications," *Archives of Computational Methods in Engineering*, vol. 29, no. 2, pp. 763–792, 2022.

[28] L. S. A. da Silva, Y. L. S. Lúcio, L. d. S. Coelho, V. C. Mariani, and R. V. Rao, "A comprehensive review on Jaya optimization algorithm," *Artificial Intelligence Review*, vol. 56, no. 5, pp. 4329–4361, 2023.

[29] J. Brownlee, *Imbalanced classification with Python: better metrics, balance skewed classes, cost-sensitive learning*. Machine Learning Mastery, 2020.